

Aberystwyth University

FFD

Ghahremani, Morteza; Liu, Yonghuai; Tiddeman, Bernard

Published in:

IEEE Transactions on Image Processing

DOI:

[10.1109/TIP.2020.3042057](https://doi.org/10.1109/TIP.2020.3042057)

Publication date:

2021

Citation for published version (APA):

Ghahremani, M., Liu, Y., & Tiddeman, B. (2021). FFD: Fast Feature Detector. *IEEE Transactions on Image Processing*, 30, 1153-1168. [9292438]. <https://doi.org/10.1109/TIP.2020.3042057>

General rights

Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk

FFD: Fast Feature Detector

Morteza Ghahremani, *Member, IEEE*, Yonghuai Liu, *Senior Member, IEEE*, and Bernard Tiddeman

Abstract—Scale-invariance, good localization and robustness to noise and distortions are the main properties that a local feature detector should possess. Most existing local feature detectors find excessive unstable feature points that increase the number of keypoints to be matched and the computational time of the matching step. In this paper, we show that robust and accurate keypoints exist in the specific scale-space domain. To this end, we first formulate the superimposition problem into a mathematical model and then derive a closed-form solution for multiscale analysis. Since the model is formulated via difference-of-Gaussian (DoG) kernels in the continuous scale-space domain, for its applicability to discrete images, we discretize it using the undecimated wavelet transform and the cubic spline function. It is proved that setting the scale-space pyramid's blurring ratio and smoothness to 2 and 0.627, respectively, facilitates the detection of reliable keypoints. Theoretically, the complexity of our method is less than 5% of that of the popular baseline Scale Invariant Feature Transform (SIFT). Extensive experimental results show the superiority of the proposed feature detector over the existing representative hand-crafted and learning techniques in accuracy and computational time. The code and supplementary materials can be found at <https://github.com/mogvision/FFD>.

Index Terms—Feature detection, difference-of-Gaussian (DoG), undecimated wavelet transform, scale-invariant, robustness.

1 INTRODUCTION

FEATURE detection is the process of extracting salient feature points from an image. The feature points could be blobs, corners or even edges [1], [2]. Depending on the application, some operations are applied to the detected feature points. Feature detection finds numerous applications in the real world such as visual localization and 3D reconstruction. A good feature detector must provide reliable interest points/keypoints that are scale-invariant, highly distinguishable, robust to noise and distortions, valid with high repeatability rate, well localized, of easy implementation and computationally fast. Over the last three decades, a large number of image local feature detectors have been proposed, in which Scale Invariant Feature Transform (SIFT) [3] is probably the most well-known technique and it actually opened a new era for image processing and computer vision. Since then, a considerable number of feature detectors have been proposed, where in most cases they followed and borrowed the concepts from SIFT like [4], [5], [6], [7], [8], [33], [34], [49].

In the literature, feature detectors can be grouped into intensity-based, multiscale and learning-based categories. Intensity-based detectors are directly applied to the grey values of images. As expected, these detectors are usually fast. The Harris corner detector and its variants [9], Features from Accelerated Segment Test (FAST) [10], Maximally Stable Extremal Regions (MSER) [11], Intensity-Based Regions (IBR) [12] and Smallest Uni-value Segment Assimilating Nucleus (SUSAN) [13] are the most representative methods in this category.

The feature detectors of the second category use scale-

space analysis. The input image is first transformed into a scale-space pyramid and then keypoints are detected. In the literature, such methods are often called multiscale feature detectors. Some representative multiscale feature detectors include SIFT, Speeded-Up Robust Feature (SURF) [4], Harris-Affine and Hessian-Affine [14], Affine SIFT (ASIFT) [5], a nonlinear scale-space method called KAZE¹ [6], Scale-Invariant Feature Detector with Error Resilience (SIFER) [7], Combination Of Shifted Filter Responses (COSFIRE) [8] and multiscale Harris corner detector (HarrisZ) [15]. Keypoints detected by the multiscale methods are usually of high accuracy, repeatability, robustness and scale-invariance. When compared to the intensity-based methods, they show better performance [19], but usually require considerably more computational time. In most applications, the feature detection step is followed by a feature description step and it is necessary to feed the descriptors with reliable keypoints, since reliable keypoints not only decrease the computational time of description but also increase the subsequent matching performance. Recently, several learnt feature detectors were developed [24], [25], [26], [28], [32], [33]. In contrast with the methods in the former two categories, the methods in the third category do not extract and analyze particular features of the images for the identification of keypoints, but automatically learn and evaluate where they are and/or how they can be described. Even though such learning-based methods have the most potential, training data limits their applicability in practice. Other interesting feature detectors can be found in [16], [27], [29], [30], [31], [36]. Comprehensive surveys on local feature detectors are provided in [17], [18].

Most of the conventional detectors cannot provide reliable keypoints and they usually fall in superimposed extrema while requiring considerable computational time. In this paper, we propose a novel multiscale feature de-

• M. Ghahremani and B. Tiddeman are with the Department of Computer Science, Aberystwyth University, UK, SY23 3DB.

E-mail: {mog9, bpt}@aber.ac.uk

• Y. Liu is with the Department of Computer Science, Edge Hill University, UK, L39 4QP.

E-mail: yonghuai.liu@edgehill.ac.uk

Manuscript received x; revised x.

1. KAZE means wind in Japanese and it stands for the nonlinear processes of the detector.

tector for computer vision applications. Firstly, while the Difference-of-Gaussian (DoG) is often used to approximate the Laplacian of Gaussian (LoG), we analyze their relations in scale normalization and excitatory regions. The analysis reveals insights into the design of a suitable DoG kernel for feature detection in the continuous scale-space domain. This kernel ensures that the approximated LoG by the DoG is scale-normalized, the blurring ratio is optimized and the DoG will not produce superimposed extreme responses for the detection of keypoints in discrete images. The proposed kernel is then discretized for effective implementation using well-structured undecimated wavelets and the spline function to form our multiscale space domain. We search for reliable blobs laid at conjunctions via analysis of the hessian matrix and an anisotropic metric. The scale-space pyramid of the proposed method does not need either upsampling or downsampling operations and thus provides good localization for the detected keypoints. Theoretically, the computational time of the proposed feature detector is about 5% that of SIFT while the keypoints detected by our detector are much more accurate and reliable than those of SIFT. Increasing reliability and reducing the computational time considerably are the main characteristics of the proposed technique. For this reason, the proposed feature detector is called fast feature detector, and for simplicity we abbreviate it as FFD.

The rest of this paper is organized as follows. In the next section, we critically review existing feature detectors. The proposed fast feature detector is detailed in Section 3. Section 4 reports and discusses the experimental results of FFD and the state-of-the-art keypoint detectors and, finally, conclusions and future work are provided in Section 5.

2 RELATED WORK

In order to critically review SIFT², two issues should be considered: (i) the framework of SIFT and (ii) the methodology behind its implementation. As discussed before, the framework of SIFT is well-established. In short, SIFT transforms an input image into a suitable scale-space, which is scale-invariant (taking this feature into the design is important as we are interested in the scale-invariant keypoints in most applications), then in the scale-space, it detects interest blobs (candidate keypoints) and refines their locations in scale-space and, finally, rejects the unstable ones. As SIFT is in favour of blobs located at conjunctions, it computes the hessian matrix for each keypoint and selects the most reliable ones using a threshold on its eigenvalues. The majority of its computational time is assigned to the construction of its Gaussian scale-space pyramid. The blurring process of SIFT is slow, and aside from its high computational cost, it produces some unreliable keypoints due to its Gaussian smoothness. Because of its scale-space, a considerable number of keypoints detected by SIFT are located over superimposed edges that lead to increase in the running time of the descriptors and decrease in matching performance subsequently, while reliable keypoints can improve the performance of the feature descriptors substantially.

2. Hereafter, for simplicity we denote the SIFT detector by SIFT, as the SIFT descriptor is not the study subject of this paper. The same notion is used for other methods.

Bay *et al.* [4] proposed a modified version of SIFT called SURF that approximates the Gaussian kernel by the integral image and Haar wavelets. While its computational time is significantly lower, approximated estimation of the Gaussian function seriously affects the localization and thus reliability of the detected keypoints. The same observation can be made in the results of BRISK [49], which is a scale-invariant version of FAST.

To address the scale smoothing of SIFT, Alcantarilla *et al.* [6] proposed ‘KAZE’. This feature detector uses a nonlinear diffusion filter to form a nonlinear scale-space, and then detects the interest points. As it uses a nonlinear filter, it is robust to noise; moreover, as there is neither up-sampling nor downsampling operation in its design, good localization is its another positive aspect. It, however, requires high computational time due to its nonlinear filter; to cope with this problem, its fast version under the name of ‘Accelerated KAZE (AKAZE)’ [20] was proposed. The computational time is reduced but still high. Roughly speaking, AKAZE needs the same computational time as SIFT. Aside from the high complexity of KAZE and its accelerated version (this is because of the estimation of nonlinear filters), the detected keypoints often fall in superimposed extrema and their reliability against distortion is low. Another improved version of KAZE is reported in [21]. Using a cosine modulated Gaussian filter [7] is another proposed method for improving the performance of SIFT. According to the reported results, this method named SIFER enhanced the repeatability of the detected keypoints, but its computational time is considerably high and it seems to be unusable in practice. The same problem can be seen with techniques in [22] and [23].

Recently, several deep learning-based feature detectors have been developed [24], [25], [26], [28], [32], [33]. They train on patch-wise/full-sized images and often provide keypoints that are robust to distortion. Even though the learnt feature detectors have a certain degree of scale invariance because of pre-training with data augmentations [32], [35], they are not inherently invariant to scale changes and their matching tends to fail in cases with a significant change in scale. In fact, data augmentation often captures well the variations in the real-world at the local level, but their effectiveness over large-scale datasets is usually difficult to predict.

Falling into superimposed extrema is the main problem of the most existing feature detectors regardless of their categories. The superimposition phenomenon is the interaction/interference between two or more adjacent edges in images whose kernel responses do not provide clear information about where these edges are. It happens in the cases that the parameters of the scale-space pyramids are not well defined. In the following section, we will show that the reliable keypoints exist only in the specific scale-space and then reconstruct the proposed multiscale pyramid based on this. This study is the first attempt to solve the superimposed extrema problem for feature detection.

3 PROPOSED FAST FEATURE DETECTOR (FFD)

Multiscale keypoint detectors generally contain two steps: scale-space pyramid construction and keypoint detection. FFD is a multiscale feature detector for finding reliable blobs

in images. We first need to design a suitable kernel for edge detection. In Section 3.1, we explore a new relation between DoG and LoG kernels. This provides a solid foundation for designing our continuous scale-space. We prove that the parameters of the scale-space, i.e. smoothness and blurring ratio, could not be tuned arbitrarily. We explore the exact relation between the above-mentioned parameters by formulating superimposition occurred during edge detection. Section 3.2 reveals that edges can be more reliably detected in the continuous scale-space if the blurring ratio and smoothness are set to 2 and 0.627, respectively. These are golden values for multiscale image processing. To the best of our knowledge, this is the first attempt to formalize and optimize the scale-space for detecting reliable edges in discrete images. We discretize our continuous scale-space using undecimated wavelet transform (UWT) and the spline function. We first review them in Section 3.3 and then use them in the FFD multiscale architecture detailed in Section 3.4. The last step of FFD, i.e. keypoints detection and refinement, is presented in Section 3.5.

3.1 Kernel Design in the Continuous Scale-Space

FFD is a blob-detector. The desired blob-detector kernel is the Laplacian of Gaussian (LoG). If a two-dimensional (2D) Gaussian function $G_\sigma(x, y)$ with width σ is defined as

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

then the scale-normalized LoG function is

$$\begin{aligned} \nabla^2 G_\sigma(x, y) &= \sigma^2 \nabla^2 G_\sigma(x, y) \\ &= \sigma^2 \left(\frac{\partial^2 G_\sigma(x, y)}{\partial x^2} + \frac{\partial^2 G_\sigma(x, y)}{\partial y^2} \right) \\ &= \frac{1}{\pi\sigma^2} \left(\frac{x^2+y^2}{2\sigma^2} - 1 \right) e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2) \end{aligned}$$

where ∇^2 denotes the Laplacian operator in 2D space. In practice, LoG is not applicable to feature detection due to its high computational complexity and noise amplification since it contains the second order derivative operations¹. Lindeberg [37] and Lowe [3] approximated an LoG function by a DoG one. They first replace time with scale in the heat diffusion

$$\frac{\partial G_\sigma(x, y)}{\partial \sigma} = \sigma \nabla^2 G_\sigma(x, y). \quad (3)$$

If the derivative of the Gaussian function is defined as

$$\frac{\partial G_\sigma(x, y)}{\partial \sigma} = \lim_{\mu\sigma \rightarrow \sigma} \frac{G_{\mu\sigma}(x, y) - G_\sigma(x, y)}{\mu\sigma - \sigma}, \quad (4)$$

then Eq. (3) can be approximated by

$$\nabla^2 G_\sigma(x, y) \approx \frac{1}{\mu - 1} (G_{\mu\sigma}(x, y) - G_\sigma(x, y)). \quad (5)$$

Here ' $G_{\mu\sigma}(x, y) - G_\sigma(x, y)$ ' denotes the DoG filter [hereafter it is denoted by $D_{\sigma,\mu}(x, y)$] and parameter μ is the ratio of two sigma values in the DoG function and is called blurring ratio. Equation (5) states that the scale-normalized LoG function can be implemented by the DoG function.

Compared to LoG, DoG is more robust to noise since it comprises two Gaussian filters that are inherently low-pass filters and thus can attenuate the side effects of noise. Moreover, the complexity of LoG is significantly reduced by DoG. However, this equation just provides an approximation for Eq. (3) and could not describe the exact relation between the scale-normalized LoG and DoG functions.

To solve this problem, we define the exact relation between the scale-normalized LoG and DoG functions as follows

$$\nabla^2 G_{\sigma_L}(x, y) = \eta(\sigma_L, \sigma, \mu) D_{\sigma,\mu}(x, y), \quad (6)$$

where σ_L denotes the sigma value of the scale-normalized LoG kernel. In the above equation, $\eta(\sigma_L, \sigma, \mu)$ is a function that makes balance between two sides of the aforementioned equation. Needless to say that $\eta(\sigma_L, \sigma, \mu)$ is approximated as ' $\frac{1}{\mu-1}$ ' in Eq. (5). In this study, we investigate its exact value.

We first assume that $\eta(\sigma_L, \sigma, \mu)$ is independent of σ_L and σ , and then check whether this assumption is true or not. With this assumption and the linearity property of Eq. (6), the excitatory regions² of the DoG and the scale-normalized LoG kernels will be identical. The excitatory region of LoG, denoted by ω_L , is obtained via setting Eq. (2) to zero:

$$\omega_L = 2\sqrt{x^2 + y^2} = 2\sqrt{2}\sigma_L. \quad (7)$$

Likewise, one can formulate the excitatory region of DoG, ω_D , as follows:

$$\omega_D = 4\mu\sigma\sqrt{\frac{\ln(\mu)}{\mu^2 - 1}}. \quad (8)$$

Since the DoG and the scale-normalized LoG functions have the identical excitatory region, we can deduce

$$\omega_L = \omega_D \Rightarrow \sigma_L = \mu\sigma\sqrt{\frac{2\ln(\mu)}{\mu^2 - 1}}. \quad (9)$$

This equation reveals the relation between the locations of zero-crossing points in the DoG and the scale-normalized LoG functions. Now we investigate their amplitudes. The aim is to make the peaks of $\nabla^2 G_\sigma(x, y)$ and $D_{\sigma,\mu}(x, y)$ identical. The peak values of both the functions are situated at the centre, i.e. $x = 0$ and $y = 0$:

$$\nabla^2 G_{\sigma_L}(0, 0) = -\frac{1}{\pi\sigma_L^2} = -\frac{\mu^2 - 1}{2\pi\mu^2\sigma^2 \ln(\mu)}, \quad (10)$$

and

$$D_{\sigma,\mu}(0, 0) = \frac{1}{2\pi\mu^2\sigma^2} - \frac{1}{2\pi\sigma^2} = -\frac{\mu^2 - 1}{2\pi\mu^2\sigma^2}. \quad (11)$$

Inserting Eqs. (10) and (11) into Eq. (6) yields

$$\eta(\sigma_L, \sigma, \mu) = \frac{\nabla^2 G_{\sigma_L}(0, 0)}{D_{\sigma,\mu}(0, 0)} = \frac{1}{\ln(\mu)}. \quad (12)$$

This equation clearly shows that $\eta(\sigma_L, \sigma, \mu)$ is independent of σ_L and σ , as claimed earlier, and it just depends on the blurring ratio. Since the blurring ratio is always fixed in the scale-space pyramid, we may conclude that a DoG function

1. In practice, the input image is smoothed before applying the LoG kernel.

2. The excitatory region denoted by ω is the area enclosed by the two zero-crossing points in the second derivative-filters [38]. See Fig. 2(a).

with sigma value of σ and blurring ratio of μ is always scale-normalized at $\mu\sigma\sqrt{\frac{2\ln(\mu)}{\mu^2-1}}$ [see Eqs. (6) and (9)]. This is an important conclusion that the DoG function is scale-normalized under any conditions. We summarize the exact relation between the normalized LoG and DoG functions in the following:

$$\nabla^2 G_{\sigma_L}(x, y) = \frac{1}{\ln(\mu)} D_{\sigma, \mu}(x, y), \quad (13)$$

where

$$\sigma_L = \mu\sigma\sqrt{\frac{2\ln(\mu)}{\mu^2-1}}. \quad (14)$$

If we seek the behaviour of the model defined in Eq. (13) for μ around 1, firstly we need to expand ' $\ln(\mu)$ ' around ' $\mu = 1$ ' via the Taylor series

$$\ln(\mu) = (\mu-1) - \frac{1}{2}(\mu-1)^2 + \dots + \frac{(-1)^{N-1}}{N}(\mu-1)^N. \quad (15)$$

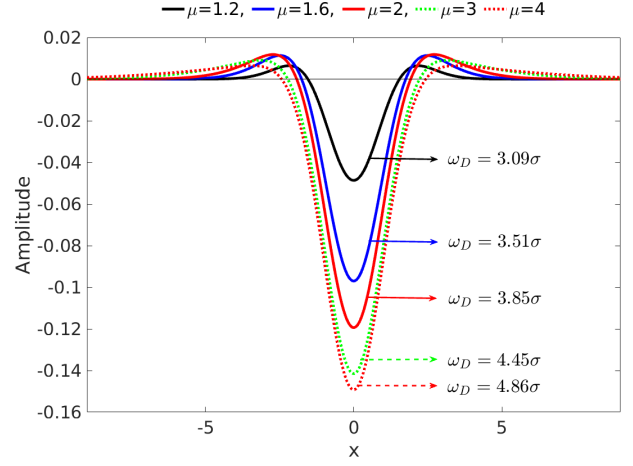
Since μ approaches 1, it is possible to approximate the Taylor series of ' $\ln(\mu)$ ' by its first term and, not surprisingly, it then yields ' $\eta(\mu) \approx \frac{1}{\mu-1}$ ', exactly as stated earlier in Eq. (5). Unlike Eq. (5) that enforces the blurring ratio to be near 1, our model shows that this parameter can actually be chosen freely in the interval of $(1, \infty)$. Now, a question arises what suitable values for μ and σ are. In the following subsection, we determine them using the superimposition concept and Eq. (14).

3.2 Determination of Blurring Ratio μ and Smoothness

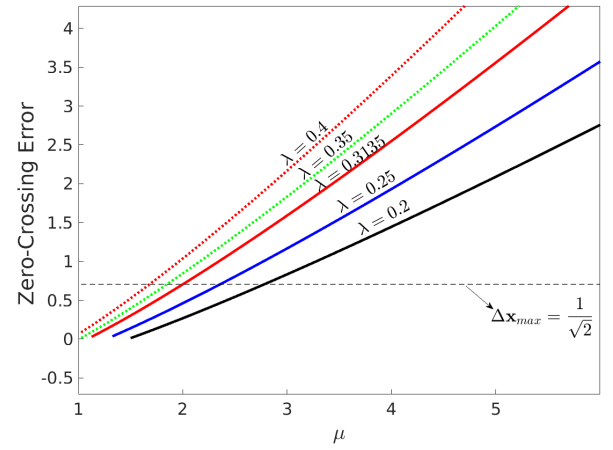
σ

Parameter μ in Eq. (13) determines the scale-ratio, controlling the ratio of two sigma values in and thus the blurring speed of the DoG function. Fig. 1(a) depicts the DoG function for different values of μ . It can be seen that the excitatory region is increased by raising the blurring ratio [recall Eq. (14)]. Determining the optimal DoG kernel has been a challenging task in image processing as the uncertainty theorem dictates in the space and the frequency domains. Marr and Hildreth [39] claimed that the suitable value of μ is 1.6. The authors in [40] mentioned that the possible values for μ could be in the range of $(1, 2]$. Lowe [3] considered ' $\mu = 2^{1/3} \approx 1.26$ ' as the best value. However, all the aforementioned values of μ are obtained under some numerical experiments and there is no formal proof for their optimal determinations. In this study, we introduce a novel framework for thoroughly analyzing features in images. We will show that the blurring ratio plays a key role in the reliability of the detected keypoints.

For simplicity, we discuss the DoG kernel in one dimension and the same results can be extended to two dimensions. When a kernel is applied to a signal, close edges may affect each other's responses. Depending on their distribution, their interaction could be destructive and amplified. Because of this interaction, superimposed edges do not show their real kernel responses so they are not reliable. Furthermore, superimposed edges may be displaced. In the image processing literature, this phenomenon is called superimposition [38]. A considerable number of detected



(a) DoG kernels for different values of μ .



(b) Zero-crossing error for different values of μ and λ .

Fig. 1. (a) Different scale-ratio values μ yield different DoG functions, and subsequently, different excitatory regions. (b) Zero-crossing error for different values of μ and λ according to Eq. (19).

blobs in the scale-space pyramid are caused by the superimposition. Here, we formulate the superimposition problem based on μ and σ and then derive a closed-form solution to their principled and optimal determination. The closeness of two adjacent edges is determined by the excitatory region ω_D of the DoG kernel. The support region is generally taken as $3\omega_D$, where 99.7% of the area under a 1D Gaussian lies in ' $3\omega_D$ ' [38]. If ' d ' denotes the distance between two adjacent edges [Fig. 2(a)], then there are three possibilities:

- $d > 3\omega_D$: The mutual influence of large-gap edges on the response to the DoG kernel is relatively weak and this can be ignored as depicted in Fig. 2(b);
- $\omega_D \leq d \leq 3\omega_D$: The mutual influence of medium-gap edges on the response to the DoG kernel is considerable [Fig. 2(c)].
- $d < \omega_D$: The mutual influence of nearby edges on the response to the DoG kernel is so strong that it cannot determine their exact locations [Fig. 2(d)].

If the width of interest region is smaller than the excitatory region of the applied kernel, i.e. $d < \omega_D$, the edges of

the given region are displaced and their zero-crossing error Δx can be computed as follows:

$$\Delta x = \sqrt{\delta x^2 + \delta y^2} = \frac{\omega_D - d}{2}, \quad (16)$$

where δx and δy are the deviation of the estimated zero-crossing point from its real value along x and y axes, respectively. The zero-crossing error increases when the difference between ω_D and d increases and in the worst case, two adjacent edges have the minimum distance from each other. If this distance is denoted by d_{min} , then

$$\Delta x_{max} = \sqrt{\delta x_{max}^2 + \delta y_{max}^2} = \frac{\omega_D - d_{min}}{2}, \quad (17)$$

where Δx_{max} is the maximum tolerable value of the zero-crossing error. The minimum distance between two adjacent edges is 1 pixel, i.e. ' $d_{min} = 1$ '. On the other hand, the maximum deviation of an extremum along x and y axes should be less than $\frac{1}{2}$ pixel, i.e. ' $\delta x_{max} = \frac{1}{2}$ ' and ' $\delta y_{max} = \frac{1}{2}$ '. This is because deviations less than $\frac{1}{2}$ pixel could be refined (this is discussed later in Eqs. (26) and (27)), otherwise there is a shift in the location of the given pixel and we should check whether it is an extremum in the new location. Considering these yields the following constraint on the excitatory region of the DoG kernel:

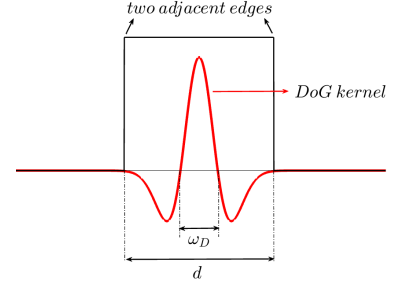
$$\Delta x_{max} \leq \frac{1}{\sqrt{2}} \Rightarrow \frac{\omega_D - 1}{2} \leq \frac{1}{\sqrt{2}}. \quad (18)$$

If we assume ' $\sigma = \lambda\mu$ ' in Eq. (14) where λ is a positive constant, then inserting it into Eq. (18) gives the following constraint on λ and μ :

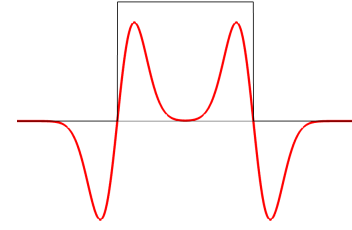
$$\frac{4\lambda\mu^2 \sqrt{\frac{\ln(\mu)}{\mu^2-1}} - 1}{2} \leq \frac{1}{\sqrt{2}}. \quad (19)$$

This constraint states that both parameters μ and σ determine the zero-crossing error. In Fig. 1(b), we depict the potential values of μ for different λ . For a fixed λ , increasing μ raises the zero-crossing error or, equivalently, the precision Δx in the space domain becomes coarse while the precision in the frequency domain (denoted by Δs) is enhanced according to the uncertainty theorem ' $\Delta x \Delta s \geq \frac{\pi}{4}$ ' [39]. As there is a trade-off between Δx and Δs , we need to select μ that satisfies Eq. (19) and at the same time yields fine precision in the frequency domain. In our experiment, the DoG kernel for μ in the range of (1, 3] has good similarity with its corresponding LoG kernel in the space domain as shown in Fig. 1(a). On the other hand, if the bandwidth of the DoG kernel is analyzed in the frequency domain, one can compute that the half-power (-3dB) bandwidth for ' $\mu = 2$ ' is about 75.3% of that at ' $\mu = 1 + 10^{-10}$ '. Hence, μ in the range of (1, 2] may provide reasonable bandwidth and we select $\mu = 2$ for its good precision both in the frequency and space domains. As will be discussed later, the space and scale locations of the keypoints will be refined by Eqs. (26) and (27) but we could not refine some other aspects of the keypoints like the superimposed extrema. By setting μ to 2, parameter λ equals 0.3135 according to Eq. (19) and this renders 0.627 for σ [see Fig. 1(b)]. In summary, σ and μ must

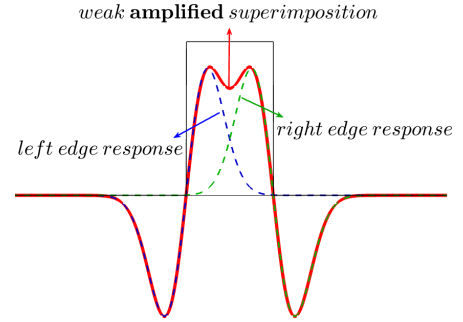
1. According to Eq. (5), σ_L is close to σ when μ approaches 1. Here, we consider ' $\mu = 1 + 10^{-10}$ ' as the closest value to 1.



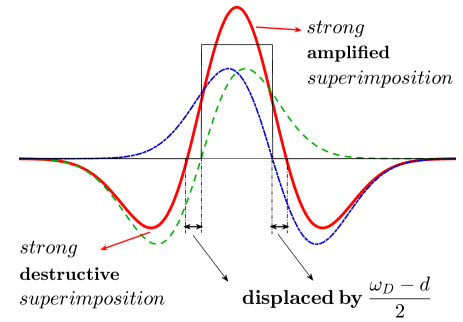
(a) Input edges and a DoG kernel



(b) $d > 3\omega_D$



(c) $\omega_D \leq d \leq 3\omega_D$



(d) $d < \omega_D$

Fig. 2. DoG kernel responses are influenced by the relation between the kernel's excitatory region ω_D and the distance d between two adjacent edges.

be carefully determined so that the responses of the DoG kernels can facilitate the separation of the nearby edges. Most conventional feature detectors have overlooked such considerations. Our analysis shows that setting σ and μ as 0.627 and 2 respectively guarantees no superimposed blobs. We thus construct our proposed multiscale space pyramid based on these golden values.

3.3 Discrete Pyramid Design by the Golden Blurring Ratio and Gaussian Kernel

Several studies [37], [47] reveal that natural images have specific properties that exist over a certain range of scales, and the possible scale-space kernel is the Gaussian function. In practical image processing, a powerful transform is the one that can provide expressive representation for the structural information of an image. The structural information is mainly edges and textures. In order to design a scale-space pyramid applicable to the discrete nature of the image using the DoG kernel with parameters ' $\sigma = 0.627$ ' and ' $\mu = 2$ ', we need to discretize the kernel. Hence, we seek a transform that can provide multiscale object representation with ' $\mu = 2$ ' and its kernel is similar to the Gaussian function in the discrete domain. We are also interested in other aspects of a good transform including translation-invariance, good localization and robustness to noise and distortions. It is clear that a robust multiscale algorithm can provide stable representation for the structural information.

Taking all the above-mentioned factors into consideration motivates us to select UWT and the spline function at the heart of our feature detector. The scale-ratio of UWT is a constant of almost 2 that makes it more suitable for our design. Moreover, UWT is an undecimated transform and it is shown that redundant transforms are robust against noise and distortions [48]. The spline kernel, on the other hand, is the approximation of the Gaussian function and is suitable for effective analysis of natural images. It is worth noting that there is a vast literature on different kernels. Haar, Daubechies, Biorthogonal, Coiflet, Symlet, Morlet, Mexican hat and different B_N -splines are probably the most applicable kernels in image processing. Likewise, there are a large number of studies on image/signal transforms. Wavelet and its numerous decimated and undecimated variants, plateaulet, ridgelet, curvelet, contourlets, bandlet, shearlet and ripplelet are the most representative transforms. Surveying all of them is out of the scope of this paper and the reader can refer to [43], [44] and references therein for more information.

Here, we briefly review the UWT with a cubic spline finite impulse response (FIR) B_3 filter bank. The undecimated wavelet transform, which is also known as stationary wavelet transform, is introduced by [41], [45]. This transform maps an image into different scale levels and then subtracts any two sequential scale/coarse images to yield the fine ones. If the kernel of UWT is the Gaussian function, then the fine scale of UWT is of the DoG functions. Instead of applying the downsampling operator to the input images, UWT upscales the kernel by a factor of 2^{j-1} , where j denotes the j th decomposition level of the image. The upsampling step is done via inserting zeros between the elements of the mother kernel and for this reason, this transform is also known as "algorithm à trous" [46]. The UWT has a redundant framework of J for J decomposition levels that makes it robust to ringing artefacts. At each scale level j , it extracts a coarse image C_j from its previous scale level ' $j - 1$ ' [43]:

$$C_j = C_{j-1} * h^{(j)}, \quad j = 1, \dots, J; \quad (20)$$

where $h^{(j)}$ denotes the kernel at scale level j and C_0 is the input image. We construct our scale-space via the above

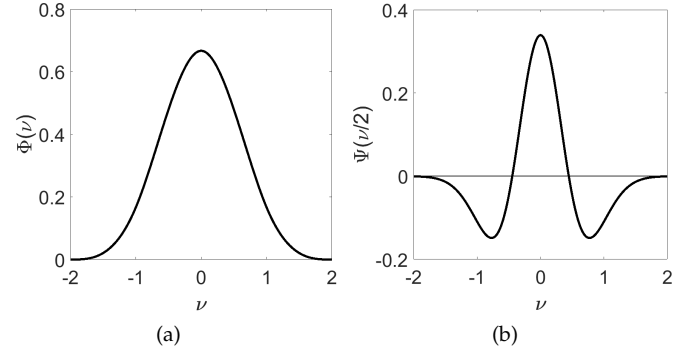


Fig. 3. (a) 1D cubic spline function Φ ; (b) 1D wavelet function Ψ .

equation and call it 'coarse scale-space pyramid'. Subtraction of any two successive layers in the coarse scale-space pyramid yields the fine one D_j as

$$D_j = C_{j-1} - C_j, \quad j = 1, \dots, J. \quad (21)$$

Similarly, the 'fine scale-space pyramid' includes all the fine images obtained via the above operation.

Since designing appropriate analysis and synthesis filter banks in image processing is a challenging task and is still open for discussion, Starck *et al.* [42] opted for the symmetric FIR B_3 filter bank. The one dimensional (1D) cubic spline function Φ [Fig. 3(a)] is defined [42] as:

$$\Phi(\nu) = \frac{1}{12} (|\nu - 2|^3 - 4|\nu - 1|^3 + 6|\nu|^3 - 4|\nu + 1|^3 + |\nu + 2|^3). \quad (22)$$

The related filter of the scaling function is $h^{(1)} = [1, 4, 6, 4, 1]/16$ and its 2D kernel is separable and obtained by convolving two 1D cubic kernels in the x and y directions respectively. Separability allows fast computation especially for large images. Other upsampled filters $h^{(j)}$, $j \in \{2, \dots, J\}$, are obtained via inserting ' $2^{j-1} - 1$ ' zeros between each pair of adjacent elements in $h^{(1)}$. The difference between two successive resolutions of the cubic function Φ yields the wavelet function Ψ [Fig. 3(b)] as:

$$\frac{1}{2}\Psi\left(\frac{\nu}{2}\right) = \Phi(\nu) - \frac{1}{2}\Phi\left(\frac{\nu}{2}\right). \quad (23)$$

3.4 FFD Multiscale Architecture

Our findings in Sections 3.1 and 3.2 concern edge detection and could be used in general image processing. Here, we utilize these concepts to build a multiscale pyramid suitable for keypoint detection.

The framework of the proposed multiscale pyramid is shown in Fig. 4. The input image is preliminarily smoothed by the Gaussian function with standard deviation of ' σ_0 ' to decrease noise and other artefacts. The possible values for σ_0 start from 0.5, where 0.5 is the minimum value to prevent significant aliasing. As will be discussed later, the value of σ_0 is set to 0.6 in our design. Next, we apply the cubic spline kernel set $\{h^{(1)}, h^{(2)}, \dots, h^{(N+2)}\}$ to the blurred image that yields ' $N + 3$ ' coarse images. According to Eq. (20), the smoothed input image with σ_0 is blurred by $h^{(1)}$ to yield the second coarse image; the resultant image is then convolved with $h^{(2)}$ to form the next coarse image, and so

forth. In fact, the coarse image in the third scale level is equivalent to the convolution of the blurred input image with kernel set $\{h^{(1)}, h^{(2)}\}$; and likewise, the fourth coarse scale-space's image is equivalent to the convolution of the blurred input image with kernel set $\{h^{(1)}, h^{(2)}, h^{(3)}\}$, and so forth. This is summarized in Table 1, where the sigma of the first kernel, σ_1 , is equal to 1.05 and this value is approximately doubled at each scale level. After organizing the coarse scale-space pyramid, the next step is to form the fine scale-space pyramid. To this end, according to Eq. (21), any two adjacent blurred images at the coarse scale-space pyramid are subtracted to yield ' $N + 2$ ' fine ones or equivalently ' N ' comparable fine ones.

As mentioned in Section 3.2, the goal is to design a DoG kernel with σ_0 and μ of 0.627 and 2, respectively. On the other hand, Table 1 states that μ is not a constant of 2 and the sigma value of the first kernel, σ_1 , is not equal to 0.627. Thus, we convolve the input image with the Gaussian function with sigma value of σ_0 in such a way that its value is around 0.627 and, simultaneously, it yields μ of almost 2. Given $\sigma_0 = \gamma\sigma_1$, where ' γ ' is a positive constant and σ_1 is 1.05 as shown in Table 1. If we arrange the scale-ratios between any two consecutive coarse images into a vector ' $\mathcal{M} = [\mu_1, \dots, \mu_{N+2}]$ ' as¹

$$\mathcal{M} = \left[\sqrt{\frac{\sigma_0^2 + \sigma_1^2}{\sigma_0^2}}, \sqrt{\frac{\sigma_0^2 + \sigma_2^2}{\sigma_0^2 + \sigma_1^2}}, \dots, \sqrt{\frac{\sigma_0^2 + \sigma_{N+1}^2}{\sigma_0^2 + \sigma_N^2}} \right], \quad (24)$$

then its length is ' $N + 2$ '. In our experiment, all the elements of vector \mathcal{M} approach 2 when γ is set to 0.57. For instance, vector \mathcal{M} for ' $N = 3$ ' is

$$\mathcal{M} \approx [2.02, 1.98, 1.99, 1.99, 1.99]. \quad (25)$$

Setting $\gamma = 0.57$ yields $\sigma_0 = 0.6$ and its corresponding smoothing filter is: $h^{(0)} = [0.002566, 0.1655, 0.6638, 0.1655, 0.002566]$.

Unlike the conventional detectors whose scale-space pyramids consist of several octaves and each octave includes some scale levels, the coarse scale-space pyramid of our feature detector contains just ' $N + 3$ ' undecimated scale levels. In fact, instead of downsampling the image, the kernel is upscaled. This feature helps us improve the localization of detected keypoints. Our studies have shown that FFD detects the majority of potential keypoints in fine images. This is because of the applied cubic spline kernel and the blurring ratio discussed in the previous sections. To better illustrate this fact, we compare the fine scale-space responses of SIFT and FFD for the 1D step function subject to 1% random Gaussian noise in Fig. 5. Parameters ' N ' in FFD and ' S ' in SIFT were set to 2, where the sigma values for SIFT and FFD are in the intervals of [1.6, 6.4] and [1.05, 9.5], respectively. An optimal detector should be able to detect all potential real edges and discard noisy or distorted ones. From the edge detection point of view, Fig. 5 shows that both the methods produce smooth responses in the jagged regions contaminated with noise while in the edge area, FFD provides much stronger responses than SIFT. Because of

1. ' $N + 3$ ' images in the coarse multiscale space pyramid yield ' $N + 2$ ' pairs of consecutive images in the fine multiscale space pyramid. When two Gaussian functions with sigma values σ_a and σ_b are convolved, the sigma value of the resultant convolution is equal to $\sqrt{\sigma_a^2 + \sigma_b^2}$.

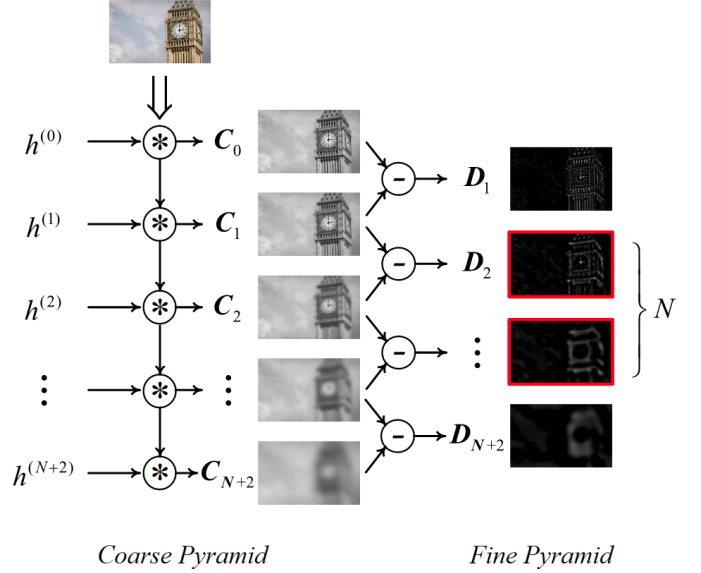


Fig. 4. The framework of the proposed multiscale pyramid. N fine images are output and used for feature detection in Section 3.5.

smoothness, SIFT ignores some potentially reliable edges, and this Achilles heel of SIFT is more observable in the images whose texture regions are not highly discriminable like those captured at night.

Good localization of the detected feature points is another important property of a good feature detector. This feature plays a pivotal role in accurately estimating parameters of interest like the fundamental matrix, homography matrix, affine transform, etc. The location of the detected edge should be as close to the true one as possible; and in the best case, the detector should return just one point for each true edge point. From Fig. 5, it can be seen that FFD responses are much closer to the true edge than SIFT's, especially in the beginning levels of scale. In the fourth level, we can see that FFD covers a large area; as mentioned earlier, FFD covers a large interval of sigma values compared to that of SIFT for the same number of scale levels. Adopting up-sampling operators and excluding downsampling operators is at the root of good localization of the keypoints detected by FFD without any ambiguity due to interpolation.

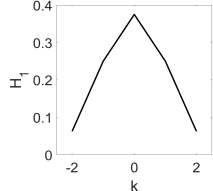
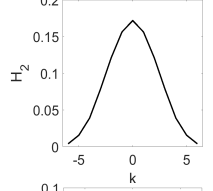
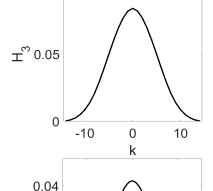
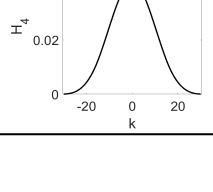
3.5 Feature Detection and Refinement

Hereafter, our task is to detect keypoints in the fine scale-space pyramid and refine their locations. Fig. 6 illustrates the keypoint detection procedure in FFD. Firstly, the candidate keypoints located at blobs in the scale-space domain are detected via a non-maximum suppression and their scale-space locations are refined [stage (I)]. To reduce false positives, these candidates are then analyzed in hessian matrix and anisotropic metric. The blobs located at conjunctions are finally taken as reliable keypoints [stage (II)]. In the following, each stage is presented in detail.

I. Extrema Detection and Refinement: Using a $3 \times 3 \times 3$ non-maximum suppression [50], the extrema blobs across space and scale are detected. Due to discretization, the extrema are often situated between pixels in the space domain and planes in the scale domain; so, we examine whether they are valid extrema and if so, where their exact scale-space

TABLE 1

The relation between kernels at different scales. h_1 is the cubic spline kernel ($h^{(1)} = [1, 4, 6, 4, 1]/16$).

Kernel	Figure	Sigma value
$H_1 = h^{(1)}$		$\sigma_1 = 1.05$
$H_2 = h^{(2)} * h^{(1)}$		$\sigma_2 = 2.32$
$H_3 = h^{(3)} * h^{(2)} * h^{(1)}$		$\sigma_3 = 4.75$
$H_4 = h^{(4)} * h^{(3)} * h^{(2)} * h^{(1)}$		$\sigma_4 = 9.5$

locations are. Similar to SIFT [3], this is done by applying the Taylor expansion to the extrema. Given that the candidate keypoint is located at $\mathbf{x} = (x, y, \sigma)$ in the k th fine image \mathbf{D}_k , $k \in \{2, \dots, N+1\}$. The quadratic Taylor expansion of the intensity $\mathbf{D}(\mathbf{x})$ is defined as

$$\mathbf{D}_k(\mathbf{x} + \Delta) = \mathbf{D}_k(\mathbf{x}) + \frac{\partial \mathbf{D}_k(\mathbf{x})}{\partial \mathbf{x}} \Delta + \frac{1}{2} \Delta^T \frac{\partial^2 \mathbf{D}_k(\mathbf{x})}{\partial \mathbf{x}^2} \Delta, \quad (26)$$

where $\Delta = (\delta x, \delta y, \delta \sigma)$ is the offset of the keypoint from the given point \mathbf{x} . Taking the derivative of Eq. (26) with respect to Δ and setting it to zero yields the offset of the candidate keypoint:

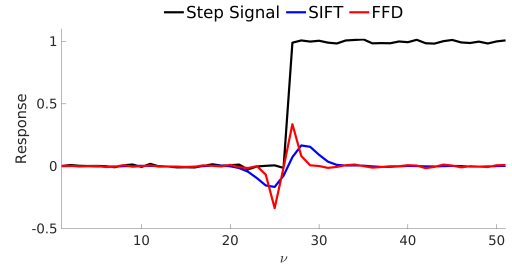
$$\hat{\Delta} = -\frac{\partial^2 \mathbf{D}_k^{-1}(\mathbf{x})}{\partial \mathbf{x}^2} \frac{\partial \mathbf{D}_k(\mathbf{x})}{\partial \mathbf{x}}. \quad (27)$$

The new location of the keypoint of interest will be $\hat{\mathbf{x}} = \mathbf{x} + \hat{\Delta}$ if each element of the offset vector $\hat{\Delta}$ is smaller than 0.5. Otherwise, the candidate keypoint is not a valid extremum and thus is discarded.

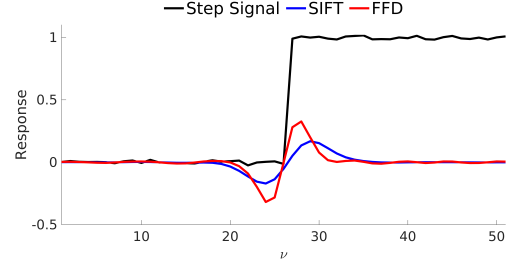
Detected blobs in the fine scale-space pyramid could be local or global extrema. Compared to global extrema, local ones have low contrast and can be removed by applying a contrast threshold, τ_{lc} , to the intensity values of the extrema, i.e. $\mathbf{D}_k(\hat{\mathbf{x}})$. The intensity values are obtained via inserting Eq. (27) into Eq. (26):

$$\mathbf{D}_k(\hat{\mathbf{x}}) = \mathbf{D}_k(\mathbf{x}) + \frac{1}{2} \frac{\partial \mathbf{D}_k(\mathbf{x})}{\partial \mathbf{x}} \hat{\Delta}. \quad (28)$$

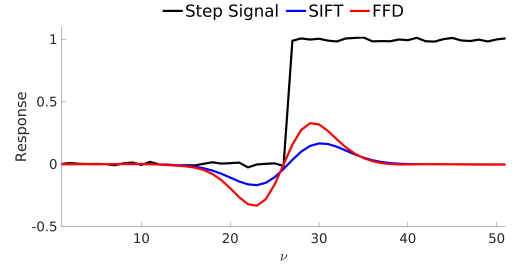
In practical image processing, the extrema with high contrast are more favourable. Bear in mind that Eq. (28) could



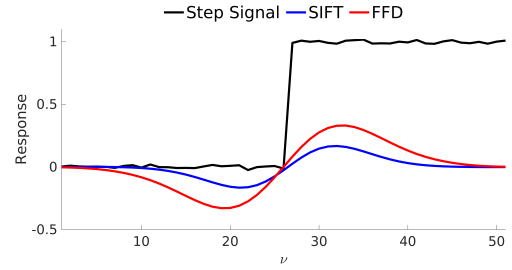
(a) 1st fine scale-space response



(b) 2nd fine scale-space response



(c) 3rd fine scale-space response



(d) 4th fine scale-space response

Fig. 5. The responses of FFD and SIFT kernels to the step function.

not remove superimposed blobs since they have large amplified values [see Fig. 2(d)]. In the case of destructive superimposition, this equation may also discard potential keypoints, increasing false negative.

II. Edge Suppression: As the detected blobs may not be reliable, the goal is to select the reliable ones, located at conjunctions. To this end, we use the anisotropy definition proposed by [51]. Let's define the tensor \mathbf{J} of the keypoint at $\mathbf{x} = (x, y, \sigma)$ as follows:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial^2 \mathbf{D}_k(\mathbf{x})}{\partial x^2} & \frac{\partial^2 \mathbf{D}_k(\mathbf{x})}{\partial x \partial y} \\ \frac{\partial^2 \mathbf{D}_k(\mathbf{x})}{\partial x \partial y} & \frac{\partial^2 \mathbf{D}_k(\mathbf{x})}{\partial y^2} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xy} \\ \mathbf{J}_{xy} & \mathbf{J}_{yy} \end{bmatrix}. \quad (29)$$

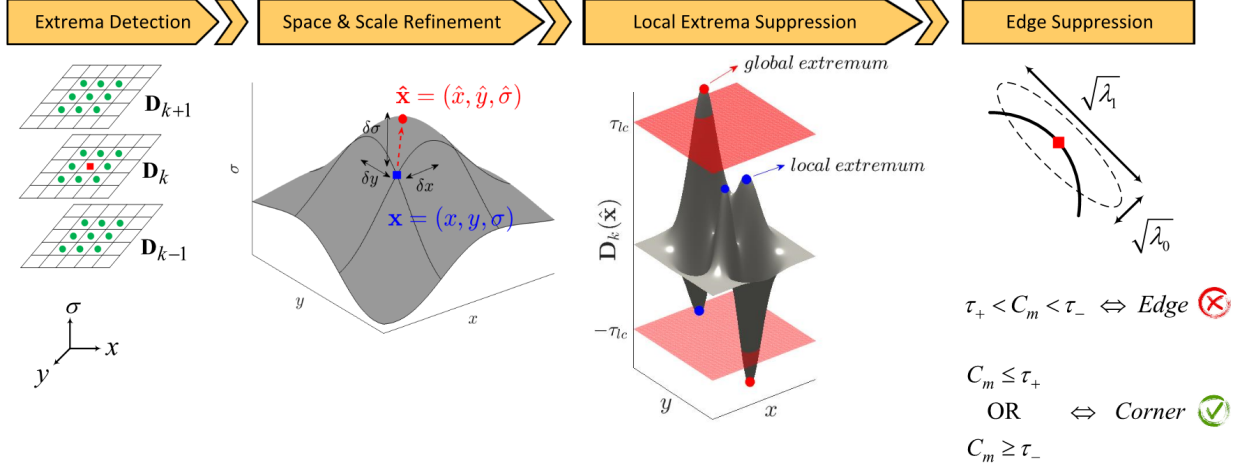


Fig. 6. The framework of salient feature detection in FFD.

Then anisotropy parameter C_m for the given pixel is defined as

$$C_m = \left(\frac{\lambda_1 - \lambda_0}{\lambda_1 + \lambda_0} \right)^c, \quad (30)$$

where λ_0 and λ_1 are the two eigenvalues of Eq. (29), and ' c ' is a positive constant. In order to avoid negative values, c is chosen as 2. For a keypoint located in conjunction, we have $\lambda_1 \approx \lambda_0$ and subsequently $C_m \approx 0$. According to the definition, they are computed from the following equation

$$\lambda_{0,1} = \frac{1}{2} \left(\mathbf{J}_{yy} + \mathbf{J}_{xx} \pm \sqrt{(\mathbf{J}_{yy} - \mathbf{J}_{xx})^2 + 4\mathbf{J}_{xy}^2} \right). \quad (31)$$

Inserting Eq. (31) into the anisotropy definition, i.e. Eq. (30), yields

$$C_m = \left(\frac{\sqrt{(\mathbf{J}_{yy} - \mathbf{J}_{xx})^2 + 4\mathbf{J}_{xy}^2}}{\mathbf{J}_{yy} + \mathbf{J}_{xx}} \right)^2 = 1 - 4 \frac{Det(\mathbf{J})}{Tr^2(\mathbf{J})}, \quad (32)$$

where $Det(\mathbf{J})$ and $Tr(\mathbf{J})$ denote the determinant and the trace of the tensor \mathbf{J} in Eq. (29):

$$Det(\mathbf{J}) = \mathbf{J}_{xx}\mathbf{J}_{yy} - \mathbf{J}_{xy}^2, \quad (33)$$

and

$$Tr(\mathbf{J}) = \mathbf{J}_{xx} + \mathbf{J}_{yy}. \quad (34)$$

C_m in Eq. (32) takes values in the interval of $[0, 1]$ as $Tr(\mathbf{J}) \geq 2\sqrt{Det(\mathbf{J})}$. If the determinant of \mathbf{J} has a large positive value, its eigenvalues are large and, subsequently, we have strong edges at multiple orientations such as conjunctions and corners. In practice, the determinant can also take negative values. To facilitate analysis, we rewrite Eq. (31) based on the determinant $Det(\mathbf{J})$ and the trace $Tr(\mathbf{J})$ of the tensor \mathbf{J} as:

$$\lambda_{0,1} = \frac{1}{2} \left(Tr(\mathbf{J}) \pm \sqrt{Tr^2(\mathbf{J}) - 4Det(\mathbf{J})} \right) = \frac{1}{2}(\alpha \pm \beta), \quad (35)$$

where $\alpha = Tr(\mathbf{J})$ and $\beta = \sqrt{Tr^2(\mathbf{J}) - 4Det(\mathbf{J})} = \sqrt{\alpha^2 - 4Det(\mathbf{J})}$. If the determinant $Det(\mathbf{J})$ takes a negative value, β is then greater than α and this means that the two eigenvalues have opposite signs. Similar to the large positive response, a large negative response also indicates

the presence of multiple edges like saddle points [19], [52]. Thus, we define two predetermined anisotropy thresholds, one for the positive determinant τ_+ and the other for the negative determinant τ_- . For each candidate keypoint, the anisotropy metric C_m is calculated via Eq. (32) and if it meets the predetermined thresholds, i.e. $C_m \leq \tau_+$ or $C_m \geq \tau_-$, then it is located at conjunction and is thus labelled as a reliable keypoint; otherwise, it is considered as an edge response and discarded.

4 EXPERIMENTAL RESULTS

We evaluate our proposed FFD against several state-of-the-art ones, including

- Multiscale methods: SIFT [3], SURF [4], KAZE [6], BRISK [49] and HarrisZ [15];
- Learning methods: TILDE [24], DNet [25], TCDet [26], LIFT [28], SuperPoint [32] and D2Net [33].

We used their implementations from OpenCV² except HarrisZ and the learning detectors. The code of HarrisZ is available in [15]. The codes and pre-trained models of the learning methods released by the authors were used here. In general, KAZE provides better results than its accelerated variant and this motivated us to compare our detector with that. It is worth noting that the computational time of AKAZE is also reported in the computational time section. The number of scales per octave for multiscale feature detectors were set to 3. In order to provide sufficient keypoints for each image, we set the detection thresholds in SIFT, KAZE and SuperPoint to 0.025, 0.0003 and 0.001, respectively; similarly, the corner detection threshold in BRISK and the keypoint detection threshold in SURF were set to 15 and 300, respectively. We used the default values for other parameters and a maximum of 10k best keypoints per image of each feature detector were selected.

For FFD³, parameters N and τ_{lc} were set to 3 and 0.05, respectively. A blob is labelled as an edge response

2. <https://opencv.org/>

3. <https://github.com/mogvision/FFD>

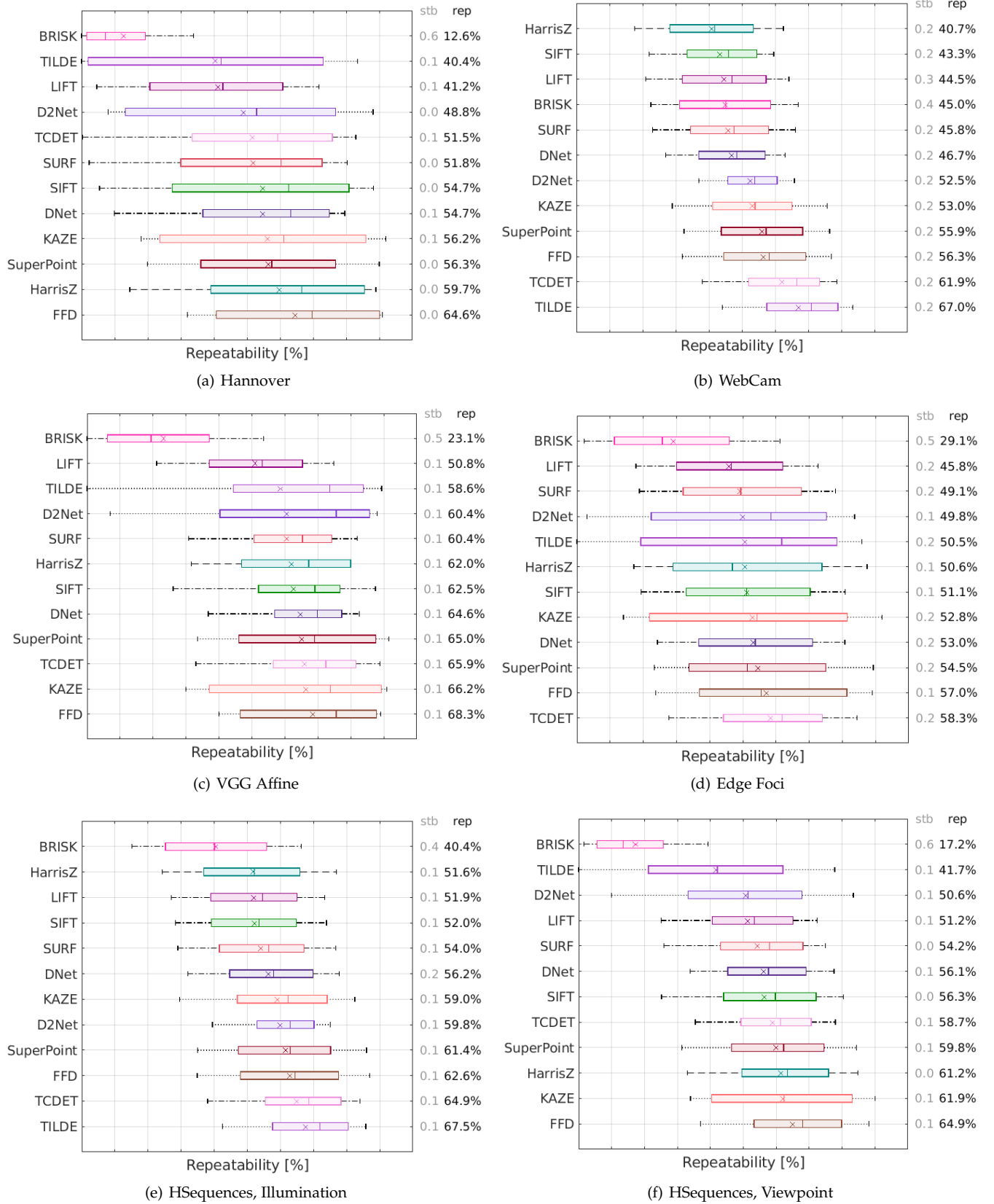


Fig. 7. Repeatability of different feature detectors over Hannover, Webcam, VGG Affine, Edge Foci, and HSequences (Illumination and Viewpoint) databases respectively.

if $0.7 \leq C_m \leq 1.5$, where the boundaries between edge and corners, i.e. τ_+ and τ_- , were set to 0.7 and 1.5, respectively. The feature points are assessed by repeatability & stability, robustness, visual localization, 3D reconstruction, golden parameter values, keypoint distribution and computational time, detailed in the following sections.

4.1 Repeatability and Stability with Homography Datasets

Here we validate the performance of the local feature detectors in repeatability and instability in the pipeline developed by Lenc and Vedaldi [58]. This pipeline was applied to several publicly available homography databases including Hannover [53], Webcam [54], VGG Affine [55], Edge Foci [56], and HSequences [57]. Mikolajczyk *et al.* [55] define the repeatability score as the fraction of keypoints that match between images with sufficient geometric overlap up to the ground-truth homography matrix. But it is revised by Lenc and Vedaldi [58] through normalization. They also introduce the instability score that quantifies the stability of the detectors across different thresholds. According to its definition, the instability of a feature detector is calculated as the standard deviation of the repeatability scores, which is then normalized by the average repeatability.

Figure 7 shows the box percentiles (first and third quartile) and the whisker percentiles of results of different feature detectors (10% and 90%). In the databases containing illumination changes, i.e. HSequences-illumination Fig. 7(e) and WebCam Fig. 4.2, the learning detectors often yield higher repeatability than the traditional ones due to their pre-training with data augmentations. In this experiment, TILDE performs well and the proposed FFD is also competitive with the learning detectors, especially in the third quartile and median values. However, TILDE is not affine invariant. In the presence of viewpoint changes, TCDET and SuperPoint outperform other learning-based feature detectors. FFD gains the highest repeatability score in three out of four viewpoint databases and TCDET wins over the remaining one. KAZE tends to have high repeatability, indicating that it is affine invariant. The stability error of most feature detectors is less than 10% while BRISK has the largest variation. If we consider the results of feature detectors over both the illumination and viewpoint sequences, it can be concluded that FFD, SuperPoint and KAZE achieve the best performance.

4.2 Robustness of Feature Detectors

Here we evaluate the robustness of the feature detectors against noise and blurring. The experiments were run over the homography databases summarised in Section 4.1, and the detected keypoints were assessed by mean average precision (mAP). The additive white Gaussian noise (WGN) with standard deviation from 0.01 to 0.2 was added to the images, even though noise can be space-variant in practical imaging. The results on the synthesized data are reported in Fig. 8, where FFD and BRISK show more resistance against noise than the other methods.

The images were also blurred by an averaging filter with various kernel sizes from 3×3 to 13×13 . The results are reported in Table 2. Since the number of detected keypoints

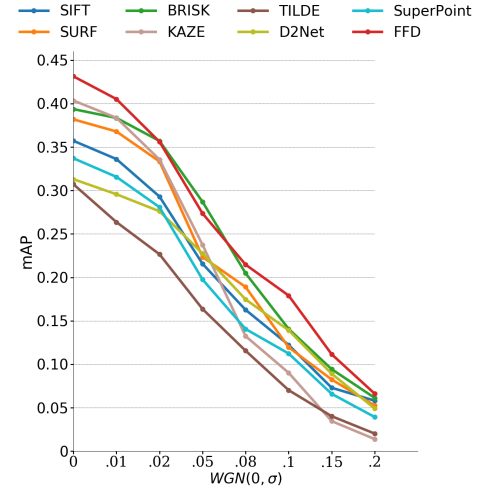


Fig. 8. mAP results of different feature detectors over the images in the homography databases corrupted by the additive white-Gaussian noise (WGN) with a standard deviation from 0.01 to 0.2.

is affected by blurring, we also reported the number of established correspondences. In terms of the number of correspondences, D2Net is less affected by blurring than the others but its mAP drops more considerably. Taking both the metrics into account, FFD, KAZE and SuperPoint are less prone to blurring.

4.3 Visual Localization

Visual localization is an important task that needs an accurate estimation of the position and orientation of the cameras. Real-world conditions like distortion, noise and day-night transitions severely affect the contents of the images and feature matching of such images is thus usually challenging. Aachen Day-Night dataset [60] contains 4,328 daytime images and 98 night-time queries. The performance of local feature descriptors is evaluated by a pre-defined visual localization pipeline⁴. The results of successfully localized images are reported with three tolerances in estimation errors of position and orientation: (0.5m, 2 deg.), (1m, 5 deg.) and (5m, 10 deg.). HardNet++ [59] was employed as the local feature descriptor for all the extracted keypoints.

The numerical results are tabulated in Table 3. The table shows that FFD achieves the best performance over all the three defined thresholds by significant margins. For strict accuracy thresholds in the estimated localization, our technique works better than all the others by as much as 2%, verifying its outstanding efficacy for localisation of the detected keypoints.

4.4 3D Reconstruction

We also evaluate the performance of the feature detectors for 3D reconstruction. According to the pipeline introduced in [61]⁵, the cameras are first calibrated in advance via Structure from Motion (SfM). Then, Multi-View Stereo (MVS) is applied to the output of SfM to obtain a dense reconstruction of the given scene. The quality of 3D models, which are

4. https://github.com/tsattler/visuallocalizationbenchmark/tree/master/local_feature_evaluation

5. <https://github.com/ahojnnes/local-feature-evaluation>

TABLE 2

The mAP (in parentheses) and the number of the established correspondences of different methods for detecting keypoints from the images blurred over different sizes of kernels.

Detector	Blur-free	3×3	5×5	7×7	9×9	11×11	13×13
SIFT	2000.8 (0.486)	1897.9 (0.481)	903.9 (0.489)	542.3 (0.473)	367.8 (0.463)	277.9 (0.451)	220.1 (0.414)
SURF	1515.8 (0.495)	1210 (0.488)	1010.6 (0.476)	767.9 (0.471)	550.3 (0.476)	401.1 (0.479)	299.9 (0.456)
BRISK	2162.6 (0.485)	1299.2 (0.468)	638.6 (0.451)	371.2 (0.499)	249.7 (0.498)	183.4 (0.445)	149.9 (0.409)
KAZE	1674 (0.513)	1511.5 (0.491)	1261.3 (0.509)	991.8 (0.518)	753.2 (0.523)	562.7 (0.512)	421.4 (0.504)
TILDE	1232.9 (0.422)	1023.1 (0.421)	872.3 (0.417)	623.3 (0.409)	491.8 (0.398)	374.9 (0.386)	277.3 (0.378)
SuperPoint	1259.1 (0.451)	1061.1 (0.453)	899.3 (0.475)	851.4 (0.463)	696.4 (0.425)	657.4 (0.396)	538.24 (0.387)
D2Net	2842.3 (0.437)	2484.1 (0.435)	2196 (0.421)	1858.6 (0.419)	1665.8 (0.395)	1402.6 (0.371)	1233.8 (0.365)
FFD	1610.9 (0.548)	1423.1 (0.531)	1247.5 (0.55)	912.1 (0.559)	713.8 (0.538)	587.5 (0.513)	448.9 (0.477)

TABLE 3

Evaluation results (%) of different detectors for visual localization on the Aachen dataset. The first and second best results are highlighted in boldface and with underline, respectively.

Detector	(0.5m, $\angle 2$)	(1m, $\angle 5$)	(5m, $\angle 10$)
SIFT	<u>42.9</u>	56.1	80.6
SURF	38.8	55.1	73.5
BRISK	39.8	<u>59.2</u>	77.6
HarrisZ	41.8	57.1	75.5
KAZE	40.6	53.0	74.4
LIFT	35.6	53.1	67.3
DNet	37.2	54.1	68.4
TILDE	38.8	54.1	69.4
TCDET	39.8	55.1	72.5
SuperPoint	40.8	<u>59.2</u>	78.6
D2Net	40.8	56.1	75.5
FFD	44.9	60.2	81.6

the outputs of MVS, directly depends on the accurate and complete estimation of the camera parameters in the first step, i.e. SfM. We follow the same metrics and protocols [61] for analysing the 3D models. According to this paper, the SfM and MVS analyses are made via COLMAP [62] and the metrics used are *the number of registered images, mean reprojection error, the number of observations, the number of inlier pairs and matches, mean track length, reconstructed sparse points and reconstructed dense points*. The datasets employed here are Fountain, Herzjesu, Madrid Metropolis, Gendarmenmarkt and Tower of London. Exhaustive image matching was employed for all the datasets and they do not need image retrieval. Similar to the previous section, keypoints were detected by different feature detectors and then described by the HardNet++ descriptor. According to the pipeline [61], the mutual nearest neighbours algorithm was employed for matching features.

The quantitative results are reported in Table 4 and Fig. 9. We reported the results of just six existing methods that gained the best performance in the previous sections. For the two smaller datasets i.e. Fountain and Herzjesu, which are relatively easy benchmarks due to the structured camera setup with high overlap, FFD performs better than the existing feature detectors in terms of the number of observations, the number of inlier matches and the number of sparse points. In the larger-scale datasets i.e. Madrid Metropolis, Gendarmenmarkt and Tower of London, which are more challenging for 3D reconstruction due to large

variations in illumination and viewpoint, FFD performs best among all the feature detectors, both in terms of sparse and dense reconstruction results. Our technique consistently produces the most complete sparse reconstruction results in terms of the number of registered images and inlier pairs, resulting in the dense models including the most points because of accurate camera pose estimation.

According to Fig. 9, FFD generally performs on par with or better than the existing techniques in terms of mean track length. The mean reprojection error shows that the multiscale techniques generally perform better than the learning ones. The localization errors of the proposed FFD are the lowest, indicating the highest precision of its detected keypoints. These results are consistent with those reported in the previous section where FFD gained the highest performance in terms of localisation accuracy.

4.5 Golden parameter values

In this section, we carry out an ablation study about whether the golden values of the parameters σ_0 and μ in the proposed FFD are optimal. To this end, we reported its results about the mAP and the number of correspondences of the detected keypoints with different values of σ_0 and μ over the Illumination and Viewpoint datasets of the HSe-quence benchmark. The experimental results are presented in Fig. 10. The figure shows that for a fixed μ , the mAP of the detected keypoints is improved by increasing σ_0 from 0.5 to 0.65 but further increasing it causes a serious decline in the number of the keypoints detected without enhancing their mAP. Likewise, for a fixed σ_0 , more deviation from $\mu = 2$ results in a lower mAP and a smaller number of the correspondences. In short, if we take both these metrics into account, ' μ around 2' and ' $0.55 \leq \sigma_0 \leq 0.65$ ' give stabler results.

4.6 Distribution of Keypoints Across Scales

Fig. 11 reports the distribution of the keypoints per scale to the total number of those detected by SIFT and FFD. The distribution of the keypoints for SIFT [Fig. 11(a)] is more even across the scales while FFD detects the majority of the keypoints at its first two scale levels (about three-quarters). There are less than one-sixth of the detected keypoints over the next scale and just less than 10% of the whole keypoints are located over the other scales. From the running time perspective, this trend is more favourable as ' $N = 3$ ' assures that the majority of reliable keypoints can be detected in

TABLE 4
Evaluation results of different feature detectors for the 3D reconstruction benchmark.

Dataset (# Images)	Descriptor	# Registered	# Observations	# Inlier Pairs	# Inlier Matches	# Sparse Points	# Dense Points
Herzjesu (8)	<i>SIFT</i>	8	38K	28	46K	11K	<u>244K</u>
	<i>BRISK</i>	8	39K	28	38K	12K	239K
	<i>KAZE</i>	8	41K	28	43K	13K	243K
	<i>TILDE</i>	8	72K	28	<u>103K</u>	19K	240K
	<i>SuperPoint</i>	8	66K	28	86K	18K	242K
	<i>D2Net</i>	8	<u>83K</u>	28	91K	<u>24K</u>	245K
	<i>FFD</i>	8	86K	28	118K	26K	245K
	<i>SIFT</i>	11	81K	55	118K	20K	<u>307K</u>
	<i>BRISK</i>	11	75K	55	81K	21K	304K
	<i>KAZE</i>	11	67K	55	75K	20K	304K
Fountain (11)	<i>TILDE</i>	11	101K	55	<u>169K</u>	24K	306K
	<i>SuperPoint</i>	11	103K	55	155K	26K	305K
	<i>D2Net</i>	11	<u>127K</u>	55	155K	<u>33K</u>	306K
	<i>FFD</i>	11	166K	55	283K	38K	308K
Madrid Metropolis (1,344)	<i>SIFT</i>	743	1.26M	896K	68.8M	251K	1.18M
	<i>BRISK</i>	731	1.19M	897K	64.7M	237K	1.16M
	<i>KAZE</i>	<u>784</u>	1.33M	<u>898K</u>	<u>70M</u>	<u>274K</u>	<u>1.31M</u>
	<i>TILDE</i>	635	696K	887K	48.2M	164K	1.05M
	<i>SuperPoint</i>	723	867K	897K	56.9M	173K	1.15M
	<i>D2Net</i>	758	1.52M	<u>898K</u>	66.2M	264K	1.26M
	<i>FFD</i>	813	<u>1.43M</u>	899K	73.1M	315K	1.36M
	<i>SIFT</i>	<u>1188</u>	2.55M	1.066M	88.4M	472K	3.04M
	<i>BRISK</i>	1145	2.36M	1.065M	74.9M	412K	3.01M
	<i>KAZE</i>	1180	2.71M	1.069M	93.6M	563K	3.06M
Gendarmen- markt (1,463)	<i>TILDE</i>	1083	2.05M	1.051M	58.8M	326K	2.98M
	<i>SuperPoint</i>	1132	1.84M	<u>1.067M</u>	64.8M	356K	<u>3.14M</u>
	<i>D2Net</i>	1154	<u>2.82M</u>	<u>1.067M</u>	90.3M	<u>611K</u>	3.08M
	<i>FFD</i>	1216	2.96M	1.069M	<u>92.4M</u>	635K	3.23M
Tower of London (1,576)	<i>SIFT</i>	1126	3.19M	1.238M	113.4M	639K	2.17M
	<i>BRISK</i>	1102	2.94M	1.237M	101.2M	514K	2.09M
	<i>KAZE</i>	1068	2.75M	1.237M	110.9M	617K	2.15M
	<i>TILDE</i>	697	1.85M	1.234M	81.6M	323K	2.01M
	<i>SuperPoint</i>	824	1.63M	1.236M	74.5M	289K	2.06M
	<i>D2Net</i>	924	2.37M	1.237M	<u>114.2M</u>	547K	2.09M
	<i>FFD</i>	1151	3.56M	1.239M	117.3M	688K	2.23M

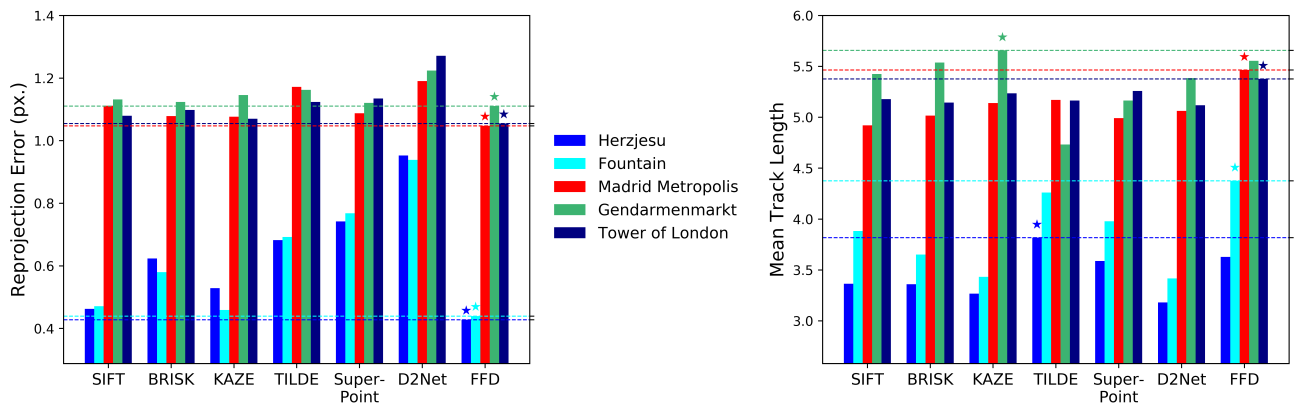


Fig. 9. The reprojection error (left) and the mean track length (right) of different feature detectors for 3D reconstruction over different datasets. The best result in each dataset is highlighted with a star.

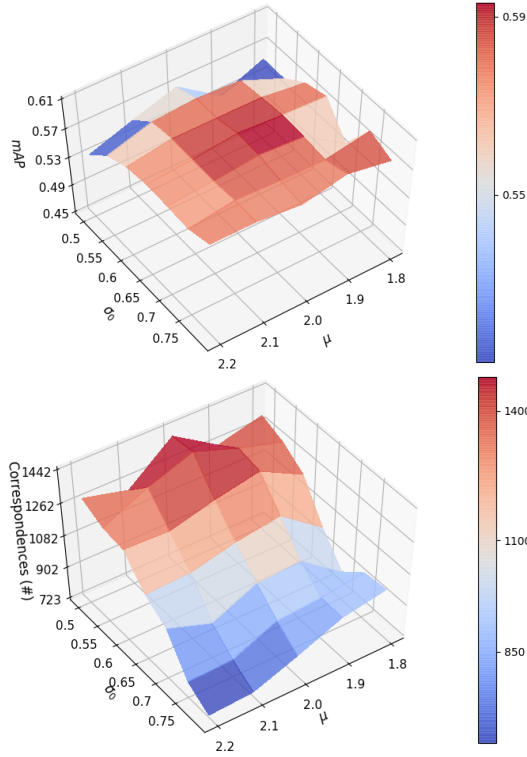


Fig. 10. The influence of parameters σ_0 and μ inside the proposed FFD on the quality (mAP and #correspondences) of the detected keypoints.

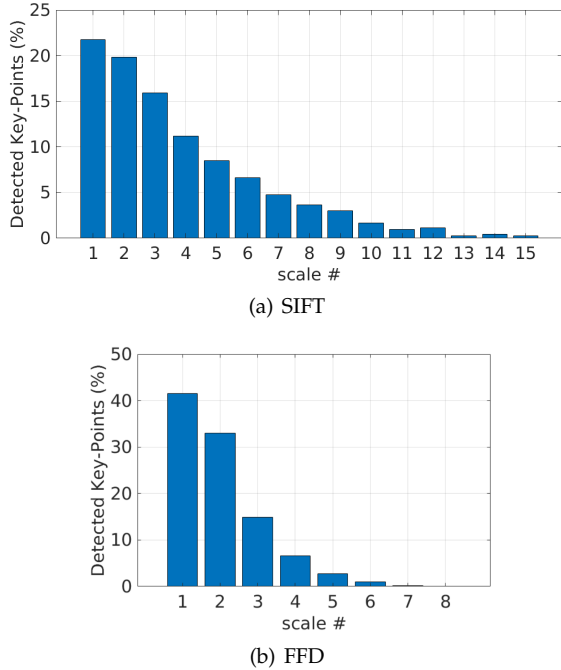


Fig. 11. The distribution of keypoints detected by SIFT and FFD across different scale levels.

the first two scale levels. Note that unlike the conventional feature detectors that split the input image into several octaves and each octave includes ' $S+3$ ' scale levels, we decompose the given image into just ' $N+3$ ' levels that includes all the octaves and scale levels.

4.7 Run Time and Computational Complexity

FFD was implemented in C++/OpenCV3.4 (without boost) and all the experiments were carried out on a 64-bits computer with Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz processors, 48 GB RAM and two Tesla P100-PCIE-16GB GPU devices.

The execution time of all the detectors as well as AKAZE is reported in Table 5. From the table, it can be seen that the computational time of HarrisZ and KAZE is high and the latter is markedly improved by its accelerated variant, i.e. AKAZE. Overall, SIFT and AKAZE need more running time while SURF and BRISK need almost one-third of that time. Although the computational time of the recent learning feature detectors like D2Net show promising results, their time-cost is still high and most of them require GPU platform. When we compare the running time of the fastest conventional feature detector, i.e. BRISK, our feature detector needs just about one-fifth of that computational time. BRISK uses the FAST feature detector at its heart that is a fast intensity-based detector. The computational time of FFD shows that it can solve the main drawback of the multiscale feature detectors, i.e. high computational cost. FFD reduces the running time of SIFT by about 95%. It is worth reporting that in FFD, 54% of the computational time is assigned to the construction of the scale-space pyramid, 25% to non-maximum suppression, and the rest to the other steps including the refinement and edge suppression. These figures for SIFT are 74%, 6% and 20%, respectively. As the majority of SIFT and FFD computational times is assigned to the pyramid construction and non-maximum suppression, we also analyze their theoretical computational complexities as follows.

The complexity of the scale-space pyramid varies with the number of scales and we consider 2 comparable scale levels where SIFT gives its largest scale-ratio, or equivalently its fastest version. The number of octaves in SIFT is set to 4, where we have one upsampling and two downsampling operations. 4 octave levels in SIFT is equivalent to setting N to 2 for FFD, as we have no upsampling operation. The length of the Gaussian filter N_f is taken as ' $7\sigma_i$ '. Both of the feature detectors use separable convolution that needs N_f multiplications and ' $N_f - 1$ ' additions. If we ignore the complexity of upsampling, SIFT needs ' $955MN$ ' operations, where M and N are the dimensions of the input image. Unlike SIFT, the kernel set in FFD is fixed at length 5 and at each scale level, the stride is changed through inserting zero between its elements; in practice, changing stride does not need arithmetic operations but at the cost of the memory accesses only. The total number of the required operations by FFD to form its fine scale-space pyramid is ' $49MN$ ', that is about 5% of the operations required by SIFT. To analyze the complexity of the non-maximum suppression step, we disregard the pixels located at borders; this step takes ' $10.625MN$ ' comparisons in SIFT and ' $2MN$ ' for FFD. Bearing these matters in mind, FFD also detects highly reliable keypoints. These remarkable characteristics show that FFD is more suitable for real-time applications.

5 CONCLUSION AND FUTURE WORK

In this study, we have proposed a novel detector, called fast feature detector (FFD). The main problems of the con-

TABLE 5

The computational time in milliseconds of different feature detectors over an image with 800×640 pixels (The average time over 20 runs is reported here).

Detector	Category	Platform	Run Time (ms)
SIFT	Multiscale	CPU	552
SURF	Multiscale	CPU	159
BRISK	Multiscale	CPU	147
HarrisZ	Multiscale	CPU	2700
KAZE	Multiscale	CPU	1500
AKAZE	Multiscale	CPU	438
DNet	Deep learning	GPU	1300
TILDE	Deep learning	CPU	12100
TCDet	Deep learning	GPU	4100
SuperPoint	Deep learning	GPU	54
D2Net	Deep learning	GPU	950
FFD	Multiscale	CPU	29

ventional feature detectors are their scale-space analysis and computational burden. We have tackled these drawbacks by analysing the relations between LoG and DoG in scale normalization and excitatory regions, where the DoG is often used to approximate LoG for the sake of computational efficiency and reduction of noise sensitivity. We proved that reliable scale-space pyramids in the continuous domain are obtained under a specific range of blurring ratio and smoothness width that are presented in Fig. 1(b). We also deduced that the blurring ratio of 2 and smoothness width of 0.627 guarantee that the resulting pyramids enable the adjacent edges in the given image to be as much separable as possible. These golden values provide valuable knowledge and insights into the design of an appropriate kernel in the continuous domain which is then discretized for its applicability to the discrete nature of the images using the undecimated wavelet transform and the cubic spline function. Experimental results and a comparative study with the state-of-the-art techniques over several publicly accessible datasets and applications show that FFD can detect more highly reliable feature points in the shortest time which makes it more suitable for real-time applications. Results have also shown that the proposed feature detector is robust to different types of distortions and variations in viewpoint and illumination. Several real-time applications like Simultaneous Localization And Map (SLAM) building and 3D phenotyping of plants require fast and robust feature detectors. Investigating the effectiveness of the proposed feature detector for such applications could be interesting.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the HPC resources provided by Supercomputing Wales (SCW) and Aberystwyth University. MG acknowledges his AberDoc and Presidents scholarships awarded by Aberystwyth University. YL is grateful to the partial funding by BBSRC and UKIERI through grants BB/R02118X/1 and DST UKIERI-2018-19-10 respectively. We thank the Associate Editor and three anonymous reviewers for their constructive comments that have improved the quality of the paper.

REFERENCES

- [1] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2005, pp. 1508-1515.
- [2] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 1994, pp. 593-600.
- [3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 91-110, Nov. 2004.
- [4] H. Bay, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346-359, Jun. 2008.
- [5] G. Yu and J. M. Morel, "ASIFT: An algorithm for fully affine invariant comparison," *Image Process. Line*, pp. 11-38, 2011.
- [6] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 214-227.
- [7] P. Mainali, G. Lafruit, Q. Yang, B. Geelen, L. V. Gool, and R. Lauwereins, "SIFER: scale-invariant feature detector with error resilience," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 172-197, 2013.
- [8] G. Azzopardi and N. Petkov, "Trainable COSFIRE filters for key-point detection and pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 490-503, May 2013.
- [9] P. Mainali, Q. Yang, G. Lafruit, L. Van Gool, and R. Lauwereins, "Robust low complexity corner detector," *IEEE Trans. Circu. Syst. Video Technol.*, vol. 21, no. 4, pp. 435-445, Apr. 2011.
- [10] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105-119, Jan. 2010.
- [11] M. Faraji, J. Shanbehzadeh, K. Nasrollahi, and T. B. Moeslund, "Extremal regions detection guided by maxima of gradient magnitude," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5401-5415, Dec. 2015.
- [12] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *Int. J. Comput. Vis.*, vol. 59, no. 1, pp. 61-85, Aug. 2004.
- [13] S.M. Smith and J.M. Brady, "SUSAN-a new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45-78, May 1997.
- [14] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63-86, Oct. 2004.
- [15] F. Bellavia, D. Tegolo, and C. Valenti, "Improving Harris corner selection strategy," *IET Comput. Vis.*, vol. 5, no. 2, pp. 86-96, Mar. 2011.
- [16] M. Tau and T. Hassner, "Dense correspondences across scenes and scales," *IEEE Trans. Image Process.*, vol. 38, no. 5, pp. 875-888, May 2016.
- [17] T. Lindeberg, "Scale selection," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Springer Publishing Company, Incorporated, Springer, pp. 701-713, 2014.
- [18] S. Wu, A. Oerlemans, E.M. Bakker, and M.S. Lew, "A comprehensive evaluation of local detectors and descriptors," *Signal Process. Imag. Communication*, vol. 59, pp. 150-167, Nov. 2017.
- [19] T. Lindeberg, "Image matching using generalized scale-space interest points," *J. Math. Imag. Vis.*, vol. 52, no. 1, pp. 3-36, May 2015.
- [20] P. F. Alcantarilla, Pablo F. and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proc. British Mach. Vis. Conf. (BMVC)*, 2013, pp. 1281-1298.
- [21] Y. Liu, C. Lan, C. Li, F. Moa, and H. Wang, "S-AKAZE: An effective point-based method for image matching," *Optik-Int. J. Light Electron Optics*, vol. 127, pp. 5670-5681, Jul. 2016.
- [22] M. A. Duval-Poo, N. Noceti, F. Odone, and E. De Vito, "Scale invariant and noise robust interest points with shearlets," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2853-2867, Jun. 2017.
- [23] S. Salti, A. Lanza and L. Di Stefano, "Keypoints from symmetries by wave propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013, pp. 2898-2905.
- [24] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "TILDE: a temporally invariant learned detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015, pp. 5279-5288.
- [25] K. Lenc and A. Vedaldi, "Learning covariant feature detectors," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 100-117.
- [26] X. Zhang, F. X. Yu, S. Karaman, and S. F. Chang, "Learning discriminative and transformation covariant local feature detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 6818-6826.

- [27] A. Mustafa, H. Kim, and A. Hilton, "MSFD: Multi-scale segmentation-based feature detection for wide-baseline scene reconstruction," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1118-1132, Mar. 2019.
- [28] K. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.
- [29] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 778-792, 2010.
- [30] M. Agrawal, K. Konolige, and M.R. Blas, "CenSurE: Center surround extremas for realtime feature detection and matching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 102-115, 2008.
- [31] W. Zhang, C. Sun, T. Breckon, and N. Alshammari, "Discrete curvature representations for noise robust image corner detection," *IEEE Trans. Image Process.*, vol. 28, no. 9, Sep. 2019.
- [32] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 224-236, 2018.
- [33] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-Net: A trainable CNN for joint detection and description of local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 8092-8101, 2019.
- [34] L. Fei, B. Zhang, Y. Xu, Z. Guo, J. Wen, and W. Jia, "Learning discriminant direction binary descriptor," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3808-3820, Aug. 2019.
- [35] P. E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 12716-12725, 2019.
- [36] L. C. Chiu, T. S. Chang, J. Y. Chen, and N. Y. Chang, "Fast SIFT design for real-time visual feature extraction," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3158-3167, Aug. 2013.
- [37] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *J. Appl. Stat.*, vol. 21, no. 2, pp. 224-270, 1994.
- [38] F. Bellavia and C. Colombo, "Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 5, pp. 651-664, Apr. 1986.
- [39] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. R. Soc. Lond.*, pp. 187-217, 1980.
- [40] T. Acharya and A. K. Ray, "Image processing: Principles and applications," *John Wiley & Sons*, 2005.
- [41] J.L. Starck, J. Fadili, and F. Murtagh, "The undecimated wavelet decomposition and its reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 297-309, Feb. 2007.
- [42] J.L. Starck and F. Murtagh, "Astronomical image and data analysis," *Springer-Verlag*, 2002.
- [43] J. L. Starck, F. Murtagh, and J. M. Fadili, "Sparse image and signal processing: Wavelets, curvelets, morphological diversity," *Cambridge university press*, Second edition, 2010.
- [44] S. Mallat, "A wavelet tour of signal processing," *Elsevier*, Third edition, 2009.
- [45] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets: Time-Frequency Methods and Phase-Space*. New York: Springer-Verlag, pp. 286-297, 1989.
- [46] M. J. Shensa, "Discrete wavelet transforms: Wedding the trous and Mallat algorithms," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2464-2482, Oct. 1992.
- [47] J.J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, pp. 363-396, 1984.
- [48] J.-L. Starck, M. Elad, and D. L. Donoho, "Redundant multiscale transforms and their application for morphological component analysis," *Adv. Imag. Electron Phys.*, vol. 132, 2004.
- [49] S. Leutenegger, M. Chli, and R.Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Con. Comput. Vis. (ICCV)*, pp. 2548-2555, 2011.
- [50] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. ICPR*, 2006.
- [51] J. Bigun and G. H. Granlund, "Optimal orientation detection of linear symmetry," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 433-438, 1987.
- [52] R. Lakemond, C. Fookes, and S. Sridharan, "Negative determinant of hessian features," in *Proc. IEEE Conf. Digital Imag. Comput. Techniques and Appl. (DICTA)*, Noosa, pp. 268-276, 2011.
- [53] K. Cordes, B. Rosenhahn, and J. Ostermann, "High-resolution feature evaluation benchmark," in *Proc. Int. Conf. Comput. Analysis Imag. Patterns*, pages 327-334. Springer, 2013.
- [54] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "TILDE: A temporally invariant learned detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015.
- [55] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, no. 1-2, pp. 43-72, Nov. 2005.
- [56] C. L. Zitnick and K. Ramnath, "Edge foci interest points," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 359-366, 2011.
- [57] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5173-5182, 2017.
- [58] K. Lenc and A. Vedaldi, "Large scale evaluation of local image feature detectors on homography datasets," In *Proc. British Mach. Vis. Conf. (BMVC)*, 2018.
- [59] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Proc. Adv. Neural Inf. Process. syst.*, 2017, pp. 4826-4837.
- [60] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, and F. Kahl, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 8601-8610.
- [61] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 1482-1491.
- [62] J. L. Schonberger and J. M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 4104-4113.



super-resolution and deep learning.

Morteza Ghahremani is working towards his PhD with a speciality in 3D computer vision at Aberystwyth University. He has published several papers in the top-ranked international journals and conference proceedings and serves as a reviewer for IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE/CVF Conference on Computer Vision and Pattern Recognition, etc. His research interests are structure from motion, 3D point cloud processing, image



Yonghuai Liu is a professor at Edge Hill University. He is an editor for several top-ranked international journals and conference proceedings in the field including Neurocomputing and Pattern Recognition Letters and International Conference on Robotics and Automation. He has published more than 190 papers in the top-ranked international journals and conference proceedings. His main research interests include 3D computer vision, image processing and machine learning.



Bernard Tiddeman is a Reader and former Head of Department in Computer Science at Aberystwyth University. He obtained his PhD from Heriot-Watt University in 1998. From 1999-2010 he worked as a researcher and then lecturer at the University of St Andrews. His research interests include: 2D and 3D facial image analysis and synthesis; computer vision for robotics; and vision and graphics applications in archaeological heritage.